



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,375	02/23/2004	Pal Takacsi-Nagy	ORACL-01389US1	8905
23910	7590	06/09/2009	EXAMINER	
FLIESLER MEYER LLP 650 CALIFORNIA STREET 14TH FLOOR SAN FRANCISCO, CA 94108			VO, TED T	
			ART UNIT	PAPER NUMBER
			2191	
			MAIL DATE	DELIVERY MODE
			06/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/784,375	Applicant(s) TAKACSI-NAGY, PAL	
	Examiner TED T. VO	Art Unit 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 April 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 50-63 and 71-81 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 50-63 and 71-81 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>04/20/09</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the amendment filed on 04/20/2009.

Claims 1-49, 64-70 are canceled. Claims 76-81 are added.

Claims 50-63, 71-81 are pending in the application.

Response to Arguments

2. This is in response to the claimed amendment and the arguments in remarks filed on 04/20/2009. The arguments are moot in view of the new ground(s) of rejection.

It should be noted that the amendment submits the recitation characterized with annotation embedding programming instructions in the markup language. Many skills in the art determine that markup languages are such as HTML, XML, etc., where XML is known as customization mark up language, are provided with programming binding. It has been well known in the art. The markup language has used the syntax `<!-- -->` , `//` , etc. In Markup language, JavaScript or VBScript is annotated within the “script” element. For example:

```
<script> <!-- - JavaScript code or VBScript code -- > </script>
```

Refer to the Peltz, it shows BPEL is in XML format. See p. 16. It should be noted that for rendering a business process using BPEL, BPEL is binding in WSDL. It is similar to HTML that binds a CSS file or JS file; or the style or JavaScript can be embedded into the HTML file using the script element within the annotation.

Art Unit: 2191

Peltz shows at least two types of code annotation: JavaDoc annotation (p. 10) that is used in this specification and java annotation tags used in BPEL as JBPEL. The script in p. 12 could be binding with markup language as mentioned above.

Therefore, the specification, which is merely to support for selecting an existing programming language and extending said existing programming language by adding workflow constructs to the existing language, reads on an script element, and later using the annotation remains reading on an customization element of an XML.

Specification

3. The use of the trademark JAVA™ has been noted in this application. See the newly amended claims. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter “computer-readable media”. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). This specification mention “workflow language”; however, Examiner fails to find how it looks like; therefore, it will cause the interpretation as the common workflow that is disclosed in Muller. In the amendment aim to make the language difference, the claim adds: *wherein the annotations to the source code, when compiled by the workflow compiler, are used*

Art Unit: 2191

to define workflow activities at the computer in accordance with the workflow definition, this limitation is not supported in the specification. The specification indeed mention a workflow compiler, but it does not describe the use of the compiler with the code the embedded within annotation.

The claim limitation should not use any feature that is not supported. It would request deleting.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 50-63, 71-81 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

As per Claims 50-56, 71-12, 76-78: While the specification mention very brief (@para:[0040]) which cannot support the recitation, “*wherein the annotations to the source code, when compiled by the workflow compiler, are used to define workflow activities at the computer in accordance with the workflow definition*”

Art Unit: 2191

As per Claims 57-63, 73-74, 79-81, and 75: See the rationale above for the similar recitation in the claims.

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 50-63, 71-81 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per Claims 50-56, 71-12, 76-78: The recitation, “*wherein the annotations to the source code, when compiled by the workflow compiler, are used to define workflow activities at the computer in accordance with the workflow definition*” that does not support in the specification renders the claim indefinite. Interpretation is that the annotations to the source code are used to define workflow activities at the computer in accordance with the workflow definition.

As per Claims 57-63, 73, 79-81, and 75: See the rationale above for the similar recitation in the claims.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A person shall be entitled to a patent unless –

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 50-63, 71-81 are rejected under 35 U.S.C. 103(a) as being unpatentable Peltz, “Web Service Orchestration”, HP, 1-2003, in view of Muller, “Event-Oriented Dynamic Adaptation of Workflows” Model, Architecture, and Implementation”.

As per claim 50: As noted that the claimed language sets its scope on workflow language, but utilizes the insertion of a program code that is usually binding in some sort of markup language.

In light of the specification, based on the standardization set forth by the XML format and browsing support; firstly, Muller discloses, a workflow definition tool for utilizing a workflow language (i.e. application programs and its application data used as a workflow engine for runtime (p. 5, and Figure 1-2)). Muller’s tool creates a source file (i.e. a box “workflow definition”) with workflow constructs to reference to the workflow language and workflow model (See Figure 1-2), which are acting as a workflow execution engine. The workflow language in general is a procedural language such as C++ or Java (p.28), or functional languages LIPS or logic

Art Unit: 2191

language Prolog. Secondly, Peltz discloses some workflow languages that use markup format using annotating to bind certain programming code such as Java.

Therefore the Workflow system of Muller discloses:

A system for utilizing a workflow language, comprising:

a computer including a processing device and a workflow compiler operating thereon (Every computer has a CPU/memory);

a program source file stored on a computer readable medium (See Figure 1-2, p. 5, A

Workflow Definition Tool generates Workflow Definition), *wherein the program source file*

includes a source code and classes therein; workflow definition created using a workflow

language that is specified in the form of annotations to the source code and classes, wherein

the annotations to the source code, when compiled by the workflow compiler, are used to

define workflow activities at the computer in accordance with the workflow definition, and

said workflow language extends the source code with a plurality of workflow constructs (See

Peltz for BPEL, and using JPBEL), *including constructs for defining parallel processing of a*

workflow and separate workflow branches therein, and wherein the workflow definition

further includes a construct to terminate the parallel processing of the workflow when certain

conditions are met; and object code executed by the processing device, the object code

configured to use a workflow program according to said workflow definition, including

processing, the plurality of workflow constructs as defined by the annotations to the source

code, to activate a workflow, including creating separate workflow processes corresponding to

the separate workflow branches (See Peltz for BPEL, and using JPBEL),

activate each of the separate workflow processes, in accordance to the annotations to the

Art Unit: 2191

source code, to subsequently generate activities at the computer as defined by each workflow branch (Figure 5-11, p. 130), **and**
determine when the certain conditions specified in the source file have occurred and then terminating the parallel processing of the workflow (See Figure 5-14 and sec. 5.4.4.3).

Muller address a generic application program of a workflow, Muller does not explicitly address, *wherein the program source file includes a source code and classes therein; workflow definition created using a workflow language that is specified in the form of annotations to the source code and classes, wherein the annotations to the source code, when compiled by the workflow compiler, are used to define workflow activities at the computer in accordance with the workflow definition, and said workflow language extends the source code with a plurality of workflow constructs.*

Peltz discloses “workflow language”, such as WSFL (i.e. Web Services Flow Language, See its “early work”, p. 5) or BPEL (p.5: BPEL4WS) that used for web service using markup language, where WSFL or BPEL includes Java (See its first paragraph, p. 11), i.e. program source file that includes source code and classes therein. This type of workflow languages can support the binding with program code embedded under annotation (Note: such as script element in HTML using the annotation for JavaScript or VBScript, etc.). See the mention of JavaDoc annotation tags (See its p. 10).

It is obvious to the ordinary in the art at the time to combine the teaching of Muller, that sets the purpose for using workflow in business process and the features embedding code in some workflow languages used in web services, as for that: some of workflow languages which are used as markup cannot be executed but rendering. They are in the XML format. Therefore,

Art Unit: 2191

for conforming to a standard of original workflow definition that uses high level languages to process activities as disclosed in Muller, it requires executable instructions embedded in, and use of annotation in Peltz when extend the type of workflow language to web, and thus will yield predictable results.

As per claim 51: Incorporated to the rejection of Claim 50, with regard to ***The system of claim 50, wherein the workflow definition is invoked by a executing a software application*** (See Peltz: “workflow definition is invoked using WSFL” (See its “early work”, p. 5), where WSFL includes Java (See its first paragraph, p. 11), or JavaDoc annotation tags (See its p. 10)).

As per claim 52: Incorporated to the rejection of Claim 50, with regard to ***The system of claim 50, wherein the plurality of workflow definition constructs are provided as XML commands*** (Muller: See p. 314, 5. rule definitions in the source file include XML –Integration) ***that are then used as annotations to the source code and classes*** (See Peltz: p. 10, note XML provides customization elements).

Peltz discloses workflow definition files are used for web service (XML-Based Language), and constructs in the files are XML-commands (See its listings, e.g. p. 7, p.8, p.17, etc).

It is obvious to the ordinary in the art at the time to combine or provide XML-commands because XML provides customized elements used for Web-service. Furthermore, using XML-commands in a workflow definition file yields the same and predictable results, and it is obvious for every ordinary in the art to include it in the teaching of Muller for web purposes.

As per claims 53, 77: Regarding 53, ***The system of claim 50, further comprising a light-weight virtual machine at the computer that processes the workflow and that is enabled to, at a***

Art Unit: 2191

particular point in the workflow process, save the workflow's execution space including program stack and variable state, and, at a later point in time, revive the workflow at the same point in the workflow process using the saved program stack and variable state.

Regarding 77, *The system of claim 53, wherein the light-weight virtual machine is configured to set the workflow program in a dormant condition; and revive the dormant workflow program to its exact state before going dormant.*

It should be notice that virtual machine is only a browser using web-service. The Reference of Peltz discloses such of web service. It should be note that Java is a stack-based application program developed by Sun Microsystems,

Therefore, it is obvious to ordinary in the art for knowing that the further claim merely recites the rules, the techniques, and the principles, which have been already developed by others as requirement in the Web, and included in the process of workflows as of Muller or Peltz for conforming to the requirements of the Web.

As per claim 54: With regards to,

The system of claim 50, wherein the program source file is a Web Service file that includes the workflow definition constructs. Refer to Peltz: The program with workflow definition constructs disclosed in Peltz is Java Web Service file.

As per claim 55: With regards to, *The system of claim 54, wherein the workflow definition constructs of the Web Service file also references Java methods and variables for a software application running on the system and using the workflow.*

Art Unit: 2191

The workflows disclosed in Peltz is Java Web Service file using method and variables running on a computer system using workflow – For example see Figure 5, p. 10.

As per claim 56: With regards to, *The system of claim 54, wherein workflow definitions are provided as a separate Work Flow file that includes workflow definition commands, and that are invoked by the Web Service file using the workflow definition constructs, to use the workflow as defined by the Work Flow file.*

The Workflow definition(s) in Muller figure 1-2 is provided as a separate Work Flow file that includes workflow definition commands, and that are invoked by the Applications program(s) using the workflow definition constructs in the files, to use the workflow as defined by the Application Program(s) as seen in Muller Figure 1-2,

It is obvious to the ordinary in the art to include separate Java Work Flow file as seen in the reference of Peltz because it yields predictable results.

As per Claim 71: Regarding, *The system of claim 55, wherein the Web Service file includes the workflow definition constructs as a plurality of XML workflow annotations to the source code and classes defined in the Web Service file.* See Peltz: p. 10.

As per Claim 72: Regarding, *The system of claim 71, wherein the XML workflow annotations to the source code and classes define a flow logic that can then reference the methods and variables defined in the Web Service file.* See Peltz: p. 10.

As per Claim 76: Regarding, *The system of claim 50, wherein the workflow definition includes*

Art Unit: 2191

a looping construct with ordering of messages received defined by a second language added to an existing programming language;

a program written using said workflow language; and

a workflow container to handle ordering of messages received in a program.

It is programming instruction per se, where any user **fre**s to use in a program, and elements in XML is standardized to provide. The looping construct is interpreted as a customized element of XML, where Muller shows it in p.314, sec. 5. Muller shows various workflow definitions expressed in schema attributes that have the form of programming constructs (p. 53-60), such as LOOP_START, and LOOP_END (p.55). It would be used as elements of XML if necessary.

As per Claim 78: Regarding, *The system of claim 50 including*

a Java programming language as the program source file, wherein the Java programming language is extended by adding workflow constructs to said Java programming language, and wherein said extending further comprises embedding the workflow constructs defined by XML in the Java programming language.

See Peltz, refer to JBPEL.

As Per Claims 57-63, and 73-74, 79-81: The rejection of the claims is the same as of Claims 50-56, and 71-72, 76-78. Refer to the rationale as addressed in the rejection of Claims 51-56, and 71-72, 76-78 above.

As Per Claim 75: The rejection of the claim is the same as of Claim 50. Refer to the rationale as addressed in the rejection of Claim 50 above.

Conclusion

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number **571-273-8300**.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV
May 28, 2009

/Ted T. Vo/
Primary Examiner, Art Unit 2191